applied sciences

MDPI

Article

Impact of Imbalanced Datasets Preprocessing in the Performance of Associative Classifiers

Adolfo Rangel-Díaz-de-la-Vega [1], Yenny Villuendas-Rey [2,*], Cornelio Yáñez-Márquez [1,*], Oscar Camacho-Nieto [2] and Itzamá López-Yáñez [2]

[1] Centro de Investigación en Computación del Instituto Politécnico Nacional, CDMX 07700, Mexico; oflodalegnar@gmail.com (A.R.-D.d.-l.V.); coryanez@gmail.com (C.Y.-M.)

[2] Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, CDMX 07700, Mexico; yenny.villuendas@gmail.com (Y.V.-R.); ocamacho@ipn.mx (O.C.-N.); itzama@gmail.com (I.L.-Y.)

* Correspondence:Correspondence: yenny.villuendas@gmail.com (Y.V.-R.); coryanez@gmail.com (C.Y.-M.)

Received: 26 February 2020; Accepted: 9 April 2020; Published: 16 April 2020

Abstract: In this paper, an experimental study was carried out to determine the influence of imbalanced datasets preprocessing in the performance of associative classifiers, in order to find the better computational solutions to the problem of credit scoring. To do this, six undersampling algorithms, six oversampling algorithms and four hybrid algorithms were evaluated in 13 imbalanced datasets referring to credit scoring. Then, the performance of four associative classifiers was analyzed. The experiments carried out allowed us to determine which sampling algorithms had the best results, as well as their impact on the associative classifiers evaluated. Accordingly, we determine that the Hybrid Associative Classifier with Translation, the Extended Gamma Associative Classifier and the Naïve Associative Classifier do not improve their performance by using sampling algorithms for credit data balancing. On the other hand, the Smallest Normalized Difference Associative Memory classifier was beneficiated by using oversampling and hybrid algorithms.

Keywords: imbalanced datasets; associative classifiers; credit scoring

1. Introduction

Credit scoring is a two-class classification problem (to grant or not the credit to the applicant). This problem is imbalanced by nature because, in practice, more credits are granted than those that are rejected. However, the classification costs are not the same for both classes, due to the inner nature of the credit assignment [1,2]. For example, if a potential good applicant is denied credit, the financial institution loses a potential client. On the other hand, if a bad applicant is granted credit, the financial institution has losses of a monetary nature, and possibly expenses associated with legal actions that it has to take to recover the money invested.

That is why the class of greatest interest in this phenomenon is the detection of potential bad applicants, who should not be granted credit [3]. Paradoxically, this class of greatest interest is the minority class in this phenomenon, which adds complexity for those involved in finding solutions to the problem of credit scoring in the context of Computational Intelligence [4].

In the recent scientific literature, there is a wide variety of pattern classifier algorithms in a wide range of applications, including Deep Neural Networks [5,6], models that show good performance. Regarding the topic of our research, it is possible to find research papers where attempts to solve the problem of credit scoring are reported. Various supervised classification models have been used in these investigations; the use of Support Vector Machines [7–9], Artificial Neural Networks [10–12] and Classifier Ensembles [13–16], among others [17–19], stands out. Some of the experimental

comparisons made to determine the performance of the classifiers in terms of credit assignment [20–23] exhibit, in our opinion, certain problems that prevent generalizing the published results.

The main task to be solved in this paper is to successfully address these two problems [24]: on the one hand, research studies incorporate few datasets, and those datasets are not public, nor are they available for use. In addition, there are almost no common datasets in the different investigations. Additionally, in the documentary study of the state of the art carried out by the authors, it was observed that, if a research group has used a certain supervised classifier, in other investigations it is not taken into account, but used other supervised classifiers.

The No Free Lunch Theorems [25] state that there is no superiority of one classifier over others, over all datasets and all performance measures. However, recent studies point to the existence of good performance of associative classifiers in solving problems of supervised classification of the financial field [26].

It is a fact known by the scientific community that, on numerous occasions, the preprocessing of the data contributes to the improvement of the performance of certain supervised classifiers; in particular, when the datasets present imbalance between classes [27]. Several investigations have been reported in the literature that have been carried out in order to determine the impact of data preprocessing in improving solutions to the problem of granting credit [28]. In particular, the computational problem related to the selection of instances (applicants) [2] has aroused great interest in the scientific community, so that in recent years the emphasis has been placed on the study of instance selection techniques for imbalanced data [1].

In this paper, we address two challenges: the evidence that in the comparative studies reviewed [20–23], there is no consensus as to what are the best preprocessing techniques for the different classifiers in the assignment of credit; and, in addition, and as a relevant point, it is a fact that, to the best of our knowledge, there is no scientific research to assess the impact of instance sampling in the performance of associative classifiers. Addressing such justifies the conduct of this research.

The aim of this paper is to successfully attack the two problems raised in the previous paragraph. Therefore, the aim of this research consists in carrying out an extensive experimental study to assess the impact of instance selection by sampling, in the performance of associative classifiers for credit scoring.

2. Previous Works

2.1. Credit Scoring

Credit scoring is one of the main income sources of financial institutions. Therefore, not having the necessary tools for customer segmentation, can cause them to be broken, due to the high delay rate of their customers. That is why, more frequently, intelligent credit granting systems (customer segmentation) are required to ensure with high probability that the future borrower will be able to meet their credit obligations, using intelligent models that facilitate and improve their approval process.

Credit scoring [29] is called any credit evaluation system for customers that allows the risk inherent in each credit application to be automatically assessed or parameterized. This risk will depend on the solvency of the client, the type of credit, the terms, and other characteristics of each client. These characteristics will define whether each credit application is approved or rejected.

Credit scoring is, therefore, a classification problem. Given a set of observations belonging to a certain class known a priori, a set of rules is search for that allow the classification of new observations into two groups: those that with high probability they will be able to face their credit obligations, and those that, on the contrary, will fail in their credit obligations.

For this, an analysis of the applicant's personal characteristics (profession, age, heritage, gender, place of residence, and others) and the characteristics of the operation (destination of credit, percentage financed, rate, term, to mention a few) will have to be carried out, which will allow the system to induce the rules that will subsequently be applied to new applications, thus determining their classification. In any case, the credit scoring models mainly use the client information evaluated

and contained in the credit applications or in internal or external sources of information. In general, Credit scoring models assign the future borrower a score (individuals and SMEs) or a rating (Business) [29].

When credit scoring techniques are used in origination (or placement), that is, to resolve credit applications, they are known as reactive or Application Scoring models. Instead, when they are used to manage the loan portfolio they are known as proactive models or Behavioral Scoring. In the case of the models used in the placement of credit, financial institutions generally determine a cutoff point to determine which applications are accepted (for obtaining a rating higher than the cutoff) and which are not. The cut off setting does not respond to risk considerations exclusively, but depends on the percentage of benefits desired by the entity and its ability to manage the risk.

2.2. Computational Intelligence Models for Financial Applications

The Computational Intelligence algorithms have been successfully applied in various branches of science and engineering [30]. Regarding the topic of our research, as of 1968 and as a result of Beaver's studies (one of the pioneers in the investigation of bankruptcy prediction models in companies) [31], several researchers began working with multivariable models with the objective of being able to determine more precisely which companies were heading for bankruptcy and which others were not. In this context, the development of the Z-Score was proposed in the year 1968 by Altman [32] that has been applied in many companies in the financial sector. For the application of credit scoring, in recent years, several new techniques have appeared, namely: Decision Trees [33], Artificial Neural Networks [12], Support Vector Machines [9], Rough Sets [19], Deep Learning [15], and Metaheuristic algorithms [34], among others.

There are several comparative studies to assess the performance of supervised classifiers for credit scoring. Maybe the first was carried out by Srinivasan and Kim [35], comparing various methodologies and find that the Decision Trees outperform the Logistic Regression, while these yield better results than the Discriminant Analysis. In addition, they suggested that the superiority of trees is directly related to the complexity of the data under study.

Ohter interesting comparatives studies are [7,21,22,29,36,37]. In addition, recent studies point to the existence of good performance of associative classifiers in solving problems of supervised classification of the financial field [26].

2.3. Data Preprocessing for Financial Applications

One of the first analyses on instance sampling on credit scoring was the one by Greene [38]. In his research paper he addressed the issue of selecting instances for predicting credit card default, and analyzed the most common technique used in credit rating: Linear Discriminant Analysis (LDA) and provided us with alternatives to this technique.

García et al. [2] conducted an investigation to analyze the impact on the presence of noise and outliers on credit risk data and establish how to improve information through data preprocessing by filtering. In the research work of López et al. [27] a comparative study was conducted to address class imbalance through instance preprocessing techniques, cost-sensitive learning and classifier ensemble methods. In addition, they analyze the impact of the intrinsic characteristics of the data on the classification task, such as small disjoints, lack of density, overlapping and separability of classes, noise and boundaries.

Crone and Finlay [39] conducted an empirical study where they analyzed the sample size and class balance. They propose that the size sufficient to build and validate a credit scoring model is 1500 to 2000 samples for each class. Bischl et al. [1], studied different strategies for the correction of class imbalance through instance sampling, and they noted that in some cases the correction worsened the performance of the classifiers, perhaps due to the over-adjustment of the training sets.

Marqués et al. [3], in the experimental results of their experiments, showed that the use of sampling methods consistently improved the performance given by the original (imbalanced) data. In addition, they mentioned that oversampling techniques worked better than any undersampling approach. Dal Pozzolo et al. [40] analyzed when undersampling was effective for imbalanced data,

and they proposed that undersampling depended on the degree of imbalance and the non-separability of classes.

García et al. [20] explored the effects of sample types on the predictive performance of classifier ensembles for credit risk and corporate bankruptcy prediction problems. They focused on characterizing positive (risky) instances, and showed that there is a correlation between the classifier ensembles performance and the dominant type of positive instances.

In conclusion, we can affirm that, although several studies have been carried out regarding the influence of the preprocessing of financial data, none of them addressed its impact on associative classifiers. In the subsequent sections, we will address this important theme.

3. Materials and Methods

This section describes the datasets and associative classifiers used in this investigation. Special emphasis is placed on datasets related to the financial environment, as they constitute a central part of this paper (Section 3.1). Additionally, the operation of the associative classifiers addressed in this research is described in detail in Section 3.2.

3.1. Datasets

This section describes the datasets that will be used to assess the impact of preprocessing of financial data in the performance of associative classifiers. Some of these datasets are well known in the literature, in addition to being a reference, because they are widely used in many of the research work carried out so far.

The used datasets are: *Australian Credit Approval* (Australian) (https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)), *German Credit data* (German) (https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)), *Japanese Credit Approval* (Japanese) (https://archive.ics.uci.edu/ml/datasets/Credit+Approval), *Default of credit card clients* (Default credit) (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients), *Iranian* (Shared personally by Hassan Sabzevari), *Polish bankruptcy* (Polish_1 to Polish_5) (https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data), *The PAKDD 2009 dataset* (The PAKDD) (https://www.cs.purdue.edu/commugrate/data/credit_card/), *Give me some credit* (Give me credit) (https://www.kaggle.com/c/GiveMeSomeCredit/data) and *Qualitative bankruptcy data* (Qualitative) (https://archive.ics.uci.edu/ml/datasets/qualitative_bankruptcy). These datasets are very interesting, due to the fact that they include both numeric and categorical attributes, and missing data.

As a summary, Table 1 shows a description of the datasets used in this investigation. The abbreviations (Num.) and (Cat.) refer to the number of numerical and categorical attributes, respectively. IR represents the imbalance ratio of the datasets.

Table 1. Description of the datasets.

Dataset	Instances	Attributes		Missing	IR
		Num.	Cat.		
Australian	690	8	6	No	1.25
Default credit	30,000	13	10	No	3.52
German	1000	7	13	No	2.33
Give me credit	150,000	10	0	No	13.96
Iranian	1002	28	0	Yes	19.04
Japanese	690	6	9	Yes	1.21
Polish_1	7027	64	0	Yes	24.93
Polish_2	10,173	64	0	Yes	24.43
Polish_3	10,503	64	0	Yes	20.22
Polish_4	9792	64	0	Yes	18.01
Polish_5	5910	64	0	Yes	13.41

Qualitative	250	0	6	No	1.34
The PAKDD	20,000	10	9	Yes	4.12

As can be seen, 10 of the 13 datasets are imbalanced (with IR > 1.5), six of them have mixed descriptions (numerical and categorical attributes), and eight contain absences of information (missing values). It should be noted that in all cases there are only two classes.

3.2. Associative Classifiers

In this section, the associative classifiers that will be evaluated in this paper are analyzed. In each case, its operation is detailed and a brief reference is made to its main characteristics, as well as its application or not to the financial field.

3.2.1. Hybrid Associative Classifier with Translation

The Hybrid Associative Classifier with Translation (HACT or CHAT by its Spanish acronym) was proposed by Santiago-Montero as a classification model [41], and has been used successfully in the financial field [42].

The HACT has two phases: association (training), and recovery (classification). This classifier assumes that the dataset is complete (that is, there are no absences of information in the data), and that it is described only by numerical attributes. In addition, it assumes that classes are represented by consecutive integers.

Let us have a dataset, having association pairs p of the form (x^μ, y^μ), where $\mu = 1, 2, \ldots, p, x^\mu \in \mathbb{R}^n$ and $y^\mu \in \mathbb{R}^m$. Each instance x^μ is composed by n components, where x_j^μ represents the j-th component of such instance. The classes y^μ take the form of binary vectors of size m, where y_i^μ represents the i-th component of the vector.

Before starting the training or association, the HACT classifier performs an axis translation process. To do this, the average of all training instances, component by component, is calculated, and then, all instances are translated considering that average. Le $\hat{x}$ be the average of all instances. Each instance x^μ is translated as $\hat{x}^\mu \leftarrow x^\mu - \hat{x}$.

After translation, class vectors are formed. To do this, the binary y^μ vectors corresponding to each instance are formed, and the corresponding component of each vector is set to 1. For example, if you have three classes (1, 2, 3) the vectors corresponding to each of the classes would be [1, 0, 0], [0, 1, 0] and [0, 0, 1], respectively. After the instances have been moved and the classes configured, the training phase of the HACT model constructs a matrix W so that when an input pattern $\hat{x}^\mu$ is presented, the stored pattern y^μ associated with the input pattern is recovered. This process comprises two basic steps:

1. For each association $(\hat{x}^\mu, y^\mu)$ in the training set, the external product $y^\mu(\hat{x}^{\mu T})$ is completed, where $\hat{x}^{\mu T}$ is the transposed of the input vector $\hat{x}^\mu$.
2. Sum the p externals products to obtain the matrix $W = \alpha \sum_{\mu=1}^{p} y^\mu(\hat{x}^{\mu T})$, where α is a normalization parameter (usually $\alpha = 1/p$). Each component of the matrix W is defined as $w_{i,j} = \sum_{\mu=1}^{p} y_i^\mu \hat{x}_j^\mu$.

On the other hand, the classification phase of HACT consists of two phases:

1. Translate the pattern to classify o, according to the average of the training patterns, as $\hat{o} \leftarrow o - \hat{x}$.
2. Determine the components of the output vector (class) for the pattern to classify $\hat{o}$. To do so, it is considered that $y_i^o = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} w_{i,j}\hat{o}_j = \max\limits_{h=1..p}\left[\sum_{j=1}^{n} w_{h,j}\hat{o}_j\right] \\ 0 & \text{otherwise} \end{cases}$. Thus, the class k wil be returned if and only if the obtained vector has value 1 in its k-th component, and 0 in the remaining components.

Although the HACT algorithm is unable of handling qualitative data, or absence of information in the data, it has obtained good results in the financial field [42].

3.2.2. Extended Gamma Classifier

The Gamma Associative Classifier was proposed by López-Yáñez as a prediction model [43], and has been used successfully in supervised classification tasks [44]. In its original version, this algorithm was unable to handle qualitative data, or absence of information in the data.

That is why an extension of this classifier was made, to overcome these limitations [45]. This extension modifies the way in which the similarity between instances is calculated, and allows the direct application of this classification model to databases with mixed and incomplete attributes, very common in the financial field.

Let X and P be the training and testing sets, respectively, from a universe U, where each instance $x \in X, p \in P$ is described by a set of attributes $A = \{A_1, A_2, \cdots, A_m\}$; each attribute A_i has a definition domain $dom(A_i)$, which can be numeric or categorical.

As a particular case, if the value of the attribute A_i in an instance x is unknown, it is considered to be a missing data, and denoted as $x_i = '?'$. It is assumed that there is a set of classes $K = \{K_1, \ldots, K_c\}$ associated with the training instances.

The Extended Gamma Associative Classifier (EG) consists of two phases: training and classification. The training phase of this classifier begins with the storage of the training set, and includes the subsequent calculation of various parameters (Table 2).

Table 2. Empirical values for the parameters of the Gamma classifier.

Parameter	Meaning	Recommendation
w	It is the vector of weights of the attributes, which indicates the importance of each attribute.	Computed by Differential Evolution [40]
θ	It is the value that will initially take θ and indicates how different two numerical values can be and that the extended generalized similarity operator considers them similar.	$\theta = 0$ (inicial value)
ρ	It is the stop parameter and refers to the maximum value allowed to θ, which allows to continue looking for the disambiguation of patterns near the border; when θ = ϱ, the CAG will stop iterating and disambiguate the class.	$\bigvee_{j=1}^{m}\left(\bigvee_{i=1}^{p} x_j^i\right)$ if there is at least a numeric attribute. Otherwise use $\rho = 1$
ρ_0	It is the pause parameter. In this pause an evaluation of the pattern to be classified is carried out, in order to determine whether or not it belongs to the unknown class: it depends on whether the normal operation of the algorithm is continued.	$\bigwedge_{j=1}^{m}\left(\bigvee_{i=1}^{p} x_j^i\right)$ if there is at least a numeric attribute. Otherwise use $\rho_0 = 1$
u	It is the threshold to decide if the pattern to be classified belongs to the unknown class or to any of the known classes.	$u = -1$

In the classification phase, EG uses an iterative process, based on the calculation of the average similarity to each class of the instance to be classified. To analyze the similarity between the test instances and training instances, the extended generalized similarity γ_{ext} is used. After obtaining the similarities, the average of the generalized similarity of said test pattern for each class $k_l \in K$ (Equation (1)) is calculated. Let $p \in P$ be an instance to classify and let p_j be the value corresponding to the j-th attribute.

The number of instances belonging to the class k_l in the training set is given by n, and x_j^i represents the value of the j-th attribute of the i-th instance of the class k_l, and w_j represents the weight of the j-th attribute.

$$c_{k_l} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} w_j * \gamma_{ext}\left(x_j^i, p_j\right)}{n} \quad (1)$$

$$\gamma_{ext}(x_j, y_j, \theta) = \begin{cases} \gamma_{num}(x_j, y_j, \theta) & \text{if the jth attribute is numeric} \\ \gamma_{cat}(x_j, y_j) & \text{if the jth attribute is categoric} \\ \gamma_{miss}(x_j, y_j) & \text{if } x_j \text{ or } y_j \text{are missing} \end{cases}$$

where

$$\gamma_{num}(x_j, y_j, \theta) = \begin{cases} 1 & \text{if } |x_j - y_j| \leq \theta \\ 0 & \text{otherwise} \end{cases}, \quad \gamma_{cat}(x_j, y_j) = \begin{cases} 1 & \text{if } x_j = y_j \\ 0 & \text{otherwise} \end{cases} \text{ and}$$

$$\gamma_{miss}(x_j, y_j) = \begin{cases} 1 & \text{if } x_j = y_j = \text{“?”} \\ 0 & \text{otherwise} \end{cases}$$

If a single maximum is found among all the values of c_{k_l}, the process ends. If not, the values of the stop and pause parameters, as well as the value of the θ parameter, are taken into account in an iterative process. Further details of the functioning of this classifier can be found in the original paper [45].

The Extended Gamma Associative Classifier has been successfully applied in solving social problems with mixed and incomplete data (particularly in estimating the voting intentions of Mexican citizens [45]). However, in the knowledge of the authors, this classifier has not been applied to the financial field.

3.2.3. Naïve Associative Classifier

The Naïve Associative Classifier (NAC) was recently proposed to solve classification problems in the financial field [26]. This classifier surpassed several of the state of the art in this type of problems; and, in addition, it has its own methodology to estimate the weight of its attributes [46].

The NAC directly handles mixed and incomplete data, and is also transportable and transparent [26]. In its training phase, the NAC stores the training set and calculates, for each numerical attribute, the standard deviation.

Let $p \in P$ be an instance to classify and let p_j be the value corresponding to the j-th attribute. To analyze the similarity between the test instance and the training instances, two operators are used: the Mixed and Incomplete Data Similarity Operator (MIDSO) and the total similarity operator s^t (Equation (2)). After obtaining the similarities, the average of the generalized similarity of the test instance for each class k_l, denoted as $s_l(p)$ (Equation (6)) is calculated.

Again, the number of instances belonging to the class k_l in the training set is given by n, and x_j^i represents the value of the j-th attribute of the i-th instance of the class k_l, and w_j represents the weight of the j-th attribute.

$$s^t(x, y) = \sum_{i=1}^{m} w_i * MIDSO(x, y, A_i) \tag{2}$$

$$MIDSO\ (x, y, A_i) = \begin{cases} s_c(x, y, A_i)\ \ if\ A_i\ is\ categorical \\ s_n(x, y, A_i)\ \ if\ A_i\ is\ numeric \end{cases} \tag{3}$$

$$s_c(x, y, A_i) = \begin{cases} 0\ si\ ((x_i \neq y_i) \vee (x_i = '?') \vee (y_i = '?')) \\ 1 \quad otherwise \end{cases} \tag{4}$$

$$s_n(x, y, A_i) = \begin{cases} 0\ si\ ((|x_i - y_i| > \sigma_i) \vee (x_i = '?') \vee (y_i = '?')) \\ 1 \quad otherwise \end{cases} \tag{5}$$

$$s_l(p) = \frac{1}{n} \sum_{y \epsilon k_l} s^t(p, y) \tag{6}$$

If a single maximum is found among all the values of $s_l(p)$, the process ends. If not, any of the classes with maximum similarity is assigned.

Although the NAC has been successfully applied to the solution of problems in the financial field, the impact of the data imbalance and its preprocessing in its performance has not been explored.

3.2.4. Smallest Normalized Difference Associative Memory

The Smallest Normalized Difference Associative Memory classifier (SNDAM) is also a newly created classification algorithm within the associative approach. It was proposed by Ramírez-Rubio

and collaborators [47], and aims to reduce the limitations of the classic Alpha-Beta associative memories.

This classification model assumes a set of training and test data described by numerical attributes, and not having absences of information. The SNDAM model is based on two operators: the generalized alpha operator α_R and the generalized beta operator β_R, in its variants MAX ($\beta_{\check{R}}$) and MIN ($\beta_{\hat{R}}$). Let c and d be two real numbers, the generalized alpha and beta operators are:

$$\alpha_R(c,d) = c - d + 1 \tag{7}$$

$$\beta_{\check{R}}(c,d) = \begin{cases} d - |c| - 1 & \text{if } c \neq d \\ c & \text{if } c = d \end{cases} \tag{8}$$

$$\beta_{\hat{R}}(c,d) = \begin{cases} c - |d| - 1 & \text{if } c \neq d \\ d & \text{if } c = d \end{cases} \tag{9}$$

For the training of this classifier, there are two aspects: the behavior as an associative memory type MAX, or as an associative memory type MIN. Depending on this, the SNDAM training phase begins as follows:

For each instance $x \in X$ in the training set, an auto-associative matrix M_x is created using the generalized alpha operator. Each component $mx_{i,j}$, with $i,j \in [1,m]$ of such matrix is given by:

$$mx_{i,j} = \alpha_R(x_i, x_j) \tag{10}$$

Subsequently, if it is a memory type MAX, an array of associations M is created whose components $m\_max_{i,j}$ are calculated according to Equation (11). In the MIN case, the components of the association matrix are created according to Equation (12). In other words, the matrices obtained for each object of the training set are generalized in a single matrix, through the maximum and minimum operators, component by component.

$$m\_max_{i,j} = \bigvee_{x \in X} mx_{i,j} \tag{11}$$

$$m\_min_{i,j} = \bigwedge_{x \in X} mx_{i,j} \tag{12}$$

After the matrix M is constructed, the maximum values of each of the attributes are calculated, considering the objects in the training set. For each attribute A_i a value MAX_i will then be associated.

Let p be a test instance that it is wanted to be recovered from the associative memory. The recovery phase in this model will return an artificial object z, described by real attributes. In the recovery phase, there are also two types: the MAX and MIN. In the first case, to obtain the components z_max_i of the artificial object z Equation (13) is used for the recovery, while in the second Equation (14) is used.

$$z\_max_i = \bigwedge_{j=1..m} \beta_{\hat{R}}(m\_max_{i,j}, p_j) \tag{13}$$

$$z\_min_i = \bigvee_{j=1..m} \beta_{\check{R}}(m\_min_{i,j}, p_j) \tag{14}$$

Then, the normalized difference δ between the recovered instance z and each of the instances of the training set $x \in X$ is calculated. Again, there are two possibilities, Max and MIN. These differences are calculated as:

$$\delta\_max(x,z) = \sum_{i=1}^{m} \frac{|z\_max_i - x_i|}{MAX_i} \tag{15}$$

$$\delta\_min(x,z) = \sum_{i=1}^{m} \frac{|z\_min_i - x_i|}{MAX_i} \tag{16}$$

Finally, the class of the instance whose normalized difference with the instance to be classified was smaller was assigned.

This classifier has been successfully applied in solving medical problems [47]. However, in the knowledge of the authors, the SNDAM classifier has not been applied to the financial field.

3.3. Sampling Algorithms for Imbalanced Data

In a large number of papers, novel methods have been proposed to address the problem of imbalance between classes, which are classified into three groups [27]; algorithm-level approaches (in which a new algorithm is created or an existing one is modified), data-level approaches (in which data is modified in order to lessen the performance impact of the algorithms of classification when there is an imbalance in the distribution of classes), and cost-sensitive classification (which consider different costs with respect to the class distribution).

This section deals with the class balancing algorithms that will be evaluated in the present investigation. All of them belong to the data-level approach for imbalanced classification. First, we address oversampling algorithms and then, undersampling algorithms. At last, we address hybrid approaches.

It is possible to find several state-of-the-art articles [48–52] in which the preprocessing of datasets is employed to reduce the impact caused by the distribution of classes. In such research, it has been empirically demonstrated that the application of a preprocessing stage to balance the distribution of classes is usually a useful solution to improve the quality of the identification of new instances. Data preprocessing techniques for imbalanced data can be divided into three groups:

1. Oversampling algorithms. These techniques are based on the creation of synthetic instances of the minority class through replication, or creating new instances based on the existing ones.
2. Undersampling algorithms. These methods are based on the elimination of instances of the majority class.
3. Hybrid Algorithms These methods are a combination of oversampling and undersampling techniques.

Oversampling algorithms seek to match the quantities of instances in each class by oversampling minority classes. In this way, the quantity of instances in these classes will be artificially increased, making all classes have approximately the same number of objects. The techniques for selecting instances for oversampling that will be used for the comparative analysis carried out in this work are listed below (Table 3).

Table 3. Oversampling algorithms used.

Name	Acronym	Reference
Synthetic Minority Over-sampling TEchnique	SMOTE	[53]
ADAptive SYNthetic Sampling	ADASYN	[54]
Borderline-Synthetic Minority Over-sampling TEchnique	SMOTE-BL	[55]
Safe Level Synthetic Minority Over-sampling TEchnique	SMOTE-SL	[56]
Random Oversampling	ROS	[57]
Adjusting the Direction Of the synthetic Minority clasS examples	ADOMS	[58]

As mentioned earlier, undersampling algorithms seek to match the amounts of instances in each class, by sampling the majority classes [25]. Thus, objects that are considered less relevant are eliminated, so that all classes have approximately the same number of instances. Next, the undersampling algorithms evaluated in the present investigation are listed (Table 4).

Table 4. Undersampling algorithms used.

Name	Acronym	Reference
Tomek's modification of Condensed Nearest Neighbor	TL	[59]
Condensed Nearest Neighbor	CNN	[60]
Condensed Nearest Neighbor + Tomek's modification of Condensed Nearest Neighbor	CNNTL	[57]
One Side Selection	OSS	[61]
Random Undersampling	RUS	[57]
Neighborhood Cleaning Rule	NCL	[62]
Under-Sampling Based on Clustering	SBC	[63]

Hybrid algorithms use oversampling and undersampling techniques. The hybrid algorithms evaluated in the present investigation are (Table 5):

Table 5. Hybrid algorithms used.

Name	Acronym	Reference
Synthetic Minority Over-sampling Technique + Edited Nearest Neighbor	SMOTE-ENN	[57]
Synthetic Minority Over-sampling Technique + Tomek's modification of Condensed Nearest Neighbor	SMOTE-TL	[57]
Selective Preprocessing of Imbalanced Data	SPIDER	[64]
Selective Preprocessing of Imbalanced Data 2	SPIDER2	[65]

In this section, we analyzed the datasets to be used, as well as some of the most representative associative classifiers. In addition, we mentioned the sampling algorithms for class balancing we will use. Next section explains the proposed experimental methodology in order to assess the impact of the preprocessing techniques for imbalanced financial data sampling in the performance of classifiers of the associative approach.

4. Experimental Methodology

This section describes the proposed methodology to assess the impact of imbalanced data preprocessing on associative classifiers. For this, the phases of this methodology are described, as well as its adaptation to the financial environment.

The proposed methodology is organized in eight steps or stages. These stages were defined by taking into account the particularities of the financial environment, the data preprocessing algorithms, and the associative classifiers.

A general description of the first six stages is given below, which will be explained in detail in the following subsections of this paper. Figure 1 shows a graphic representation of the stages of this methodology. Stages seven (execution of the experiments) and eight (analysis of the results) will be addressed in Section 5, as they correspond to the results obtained, as well as their analysis and discussion.

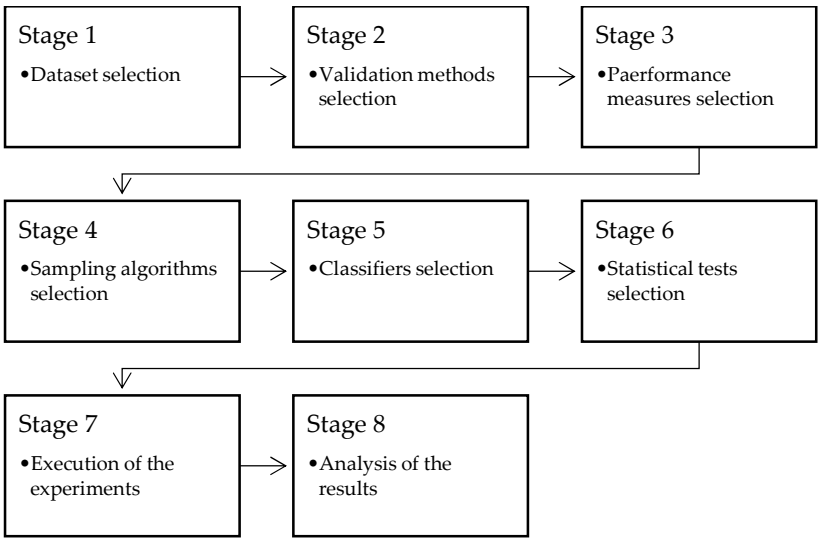

Figure 1. Stages of the proposed methodology.

4.1. Dataset Selection

As mentioned earlier, 13 datasets were considered in this investigation, all of them referring to the problem of credit scoring. The number of attributes of these datasets varies between six and 64 attributes, while the number of instances is between 250 and 150,000. On the other hand, 10 of the 13 databases are imbalanced (with IR > 1.5), six of them have mixed descriptions (numerical and categorical attributes), and eight contain absences of information. It should be noted that in all cases there are only two classes. These classes correspond to clients that do not represent a risk to banking institutions, and clients that do represent such risk.

4.2. Validation Methods Selection

There are various methods or techniques to compare the results obtained from classification algorithms. One of these techniques is cross validation. The pioneer in using cross-validation was Larson [66] in 1931. In his research work, he divided the data into two parts, a sample was used for regression and a second sample was used for prediction. In the 70's, Stone [67] and Geisser [68] formally developed the concept of cross-validation, which is a statistical method to divide a set of data into different subsets. Its application is very useful when the number of data is relatively small due to the complexity or even impossibility of obtaining it. There are different variants of validation, among which are:

Hold-Out: the complete set of data is taken and divided into two subsets, the first one dedicated to the training phase and the second to the test phase. The partition of the data is done by taking random elements.

K-fold cross validation: the data set is divided into K partitions that give K mutually exclusive subsets. That is, one of the K subsets is used as a test set and the remaining sets are grouped to form the training set. The procedure is repeated K times by exchanging the test set to ensure that the K subsets have been used in the test phase.

5 × 2 cross validation: It is important to highlight the work of Dietterich [69], who compares various evaluation methods. He proposed to apply 5 × 2 cross validation, which consists of repeating a cross validation five times with K = 2. In each of the five executions, the data is randomly divided into two subsets of data, one for training and the other for testing. Each of the training partitions are taken as input of the classification algorithm and the test partitions are used to make a test of the final solution.

Considering the high imbalance of some of the datasets used (for example, Polish_1, with an IR = 24.9), it was decided to use the 5 × 2 cross validation method, to compare the results obtained in the investigation.

For this, the datasets were divided using the KEEL software [70], which allows to store, in each case, the partitions obtained. This is a clear advantage, since the different classification algorithms are compared on the same test sets, in each case.

4.3. Performance Measure Selection

The evaluation of the performance of supervised classifiers has been the subject of study since the very emergence of the classification algorithms. One of the measures first used is to consider the number of instances of the test set correctly classified, with respect to the total instances of that set. This measure is known as accuracy or correct classification rate [71].

However, this is not the only way to evaluate the performance of the classifiers. Let us consider a two-class scenario (Positive and Negative), as shown in Figure 2. In this case, a classification algorithm has four possibilities:

1 Correctly classify a positive instance (True Positive, TP)
2 Correctly classify a negative instance (True Negative, TN)
3 Incorrectly classifying a positive instance (False Negative, FN)
4 Incorrectly classifying a negative instance (False Positive, FP)

		Assigned class	
		Positive	**Negative**
Real class	**Positive**	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Figure 2. Two class confusion matrix.

It should be noted that the costs of each of the two possibilities of incorrect classification (False Positives and False Negatives) are not always the same. In the particular case of the credit scoring, if clients that represent a risk to the financial institution are considered as a positive class, and those that do not represent a risk as a negative class, it is easy to deduce that the cost of classifying a risky client (positive) as negative, it is greater than classifying a good customer (negative) as a risk customer. In these scenarios, it is sought, above all, to reduce the amount of False Negatives.

This type of situation is aggravated by considering imbalanced datasets, since standard performance measures such as accuracy are not considered adequate [27]. This is due to the bias of these measures towards the majority class, since they do not distinguish between the number of correct classifications of the different classes, which can lead to erroneous conclusions. In this investigation, we will consider the Area under the ROC curve to evaluate the performance of the classifiers, after applying data balancing algorithms.

Below, we list some of the most commonly used performance measures [71] for imbalanced dataset scenarios, considering a two-class confusion matrix, as shown in Figure 3.

Measure	Equation	Evaluates
True Positive Rate	$TPR = \frac{TP}{TP + FN}$	Ability of the classifier to identify positive instances
True Negative Rate	$TNR = \frac{TN}{FP + TN}$	Ability of the classifier to identify negative instances
Area under the ROC curve	$AUC = \frac{TPR + TNR}{2}$	Ability of the classifier to prevent incorrect classifications

Figure 3. Some performance measures for two class imbalanced classification.

4.4. Sampling Algorithms Selection

As mentioned early, there are numerous algorithms for data balancing. These algorithms are divided into three large groups: undersampling algorithms, oversampling algorithms and hybrid algorithms. In this investigation, the KEEL tool [72] was used to apply these algorithms to the different datasets.

KEEL version 3.0 contains 20 sampling algorithms. Of them, 17 were selected for the experiments. The reason for the selection is its computational efficiency. The Aglomerative Hierarchical Clustering (AHC), Hybrid Preprocessing using SMOTE and Rough Sets Theory (SMOTE-RSB) and Class Purity Maximization (CPM) algorithms were not considered, due to the impossibility of obtaining results in 24 h for some datasets.

We used a personal computer with 2GB of RAM, Windows 7 operating system, and an Intel Core i3 processor of 5th generation. Such computer was not exclusively dedicated to the experiments execution, and therefore, we cannot analyze the execution time of the algorithms under study.

As some of these algorithms do not handle mixed or incomplete data, in the cases corresponding to datasets that do contain them, it is proposed thin this research the lost values were imputed, and the categorical attributes were converted to numerical. For this, it is possible to use the KEEL software [66] in its functionalities "Concept Most Common Attribute Value" and "Min Max ranging".

4.5. Classifiers Selection

From the associative classifiers mentioned, the following algorithms were chosen for conducting the experiments: HACT, NAC, EG and SNDAM. HACT [54] and NAC [23] have been applied previously, successfully, to the financial field [26,42].

For the execution of the associative classification algorithms, we use the EPIC tool [73,74], currently under development. This tool contains the above-mentioned associative classifiers, and it is compatible with the datasets files after applying the sampling algorithms offered by KEEL, and has a visual environment that facilitates the performance of experiments. On the other hand, the EPIC tool provides a summary of the results obtained according to numerous performance measures, and also stores the classes assigned to each test instance, in each case.

4.6. Statistical Test Selection

In the proposed methodology, it is necessary to evaluate the performance of several supervised classifiers on different datasets. To establish the existence or not of differences between performances, it is necessary to carry out statistical tests. Among the statistical tests recommended for this task [75] is the Friedman test [76].

In this case, a null hypothesis H0 is defined, which states that there are no differences in the performance of the compared algorithms, and an alternative hypothesis H1, which states that there are differences in the performance of the compared algorithms.

Friedman's test consists in ordering the performances of the algorithms (from best to worst performance), replacing them with their respective rank. The best result corresponds to rank 1, the second best to rank 2, and so on. When ordering them, the existence of identical data is considered, in which case an average range is assigned. Then, the test computes the z-statistic. It is used to find the corresponding probability value p in the statistical tables, and then compare it with a significance value α. In the tests of statistical hypotheses, the p-value represents the probability of obtaining a result as extreme as one already observed, assuming that the null hypothesis is true. The lower the p-value, the more evidence exists against the truthfulness of the null hypothesis. If the p-value is less than the level of significance α, the null hypothesis is rejected and significant differences are considered to exist.

If the null hypothesis of equal performance is rejected by the Friedman test, it is necessary to apply post-hoc tests, to determine among which algorithms are the differences. Among the recommended post-hoc tests for the analysis of algorithm performance in multiple datasets, is the Holm test [77]. This test, to adjust the significance value α, uses a descending procedure. There are

automated tools for the calculation of the Friedman test, as well as for the calculation of post-hoc tests. In this paper, we use the KEEL tool [72]. As mentioned at the beginning of this section, the first six stages of the proposed methodology were detailed. Considering the above, next section will describe stages seven and eight of the proposal, corresponding to the execution of the experiments, and their analysis.

5. Experimental Results

This section describes the experiments performed. First, an analysis is carried out on the results obtained by the sampling algorithms (Section 5.1). Then, the impact of the results obtained by these algorithms in the performance of associative classifiers is evaluated (Section 5.2). Finally, the statistical analysis is made (Section 5.3).

5.1. Results of the Sampling Algorithms

As expected, the oversampling algorithms were able to perfectly balance the datasets, obtaining and imbalance ratio of one in all cases (Table 6).

Table 6. Imbalance ratio for oversampling algorithms.

Datasets	Original	ADASYN	ADOMS	SMOTE-BL	ROS	SMOTE-SL	SMOTE
Australian	1.25	1.00	1.00	1.00	1.00	1.00	1.00
Default credit	3.52	1.00	1.00	1.00	1.00	1.00	1.00
German	2.33	1.00	1.00	1.00	1.00	1.00	1.00
Give me credit	13.96	1.00	1.00	1.00	1.00	1.00	1.00
Iranian	19.00	1.00	1.00	1.00	1.00	1.00	1.00
Japanese	1.26	1.00	1.00	1.00	1.00	1.00	1.00
Polish_1	24.93	1.00	1.00	1.00	1.00	1.00	1.00
Polish_2	24.43	1.00	1.00	1.00	1.00	1.00	1.00
Polish_3	20.22	1.00	1.00	1.00	1.00	1.00	1.00
Polish_4	18.01	1.00	1.00	1.00	1.00	1.00	1.00
Polish_5	13.41	1.00	1.00	1.00	1.00	1.00	1.00
Qualitative	1.34	1.00	1.00	1.00	1.00	1.00	1.00
The PAKDD	4.12	1.00	1.00	1.00	1.00	1.00	1.00

Considering the experiments performed, we can conclude that oversampling methods tend to obtain perfect balances. However, these results are at the cost of artificially increasing the cardinality of the data sets. The next section will analyze how these results impact the performance of classifiers of the associative approach, and if this computational cost is justified in results of better performance.

Below, the results of the undersampling algorithms are offered in Table 7, for each of the datasets analyzed. The cases in which the classes were inverted are shown in italics (the minority class became the majority), and in bold the good results (those with an imbalance closer to 1).

Table 7. Imbalance ratio for undersampling algorithms.

Datasets	Original	CNN	CNNTL	NCL	OSS	RUS	SBC	TL
Australian	1.25	*3.37*	*8.59*	*1.35*	*5.50*	**1.00**	*2.22*	*1.04*
Default credit	3.52	**1.13**	*2.24*	**1.78**	*1.21*	**1.00**	-	2.69
German	2.33	*1.12*	*2.96*	**1.08**	*1.75*	**1.00**	*1.96*	**1.68**
Give me credit	13.96	*1.59*	*1.36*	10.84	*1.39*	**1.00**	*2.00*	12.95
Iranian	19.00	2.37	**1.47**	14.53	2.18	**1.00**	-	17.91
Japanese	1.26	*3.45*	*9.07*	*1.33*	*5.30*	**1.00**	*1.96*	*1.01*
Polish_1	24.93	2.53	**1.40**	21.14	2.50	**1.00**	-	23.70
Polish_2	24.43	2.62	**1.41**	20.31	2.56	**1.00**	-	22.97
Polish_3	20.22	2.58	**1.33**	16.33	2.45	**1.00**	-	18.86

Polish_4	18.01	2.38	**1.25**	14.42	2.22	**1.00**	-	16.72
Polish_5	13.41	**1.84**	**1.15**	10.23	**1.62**	**1.00**	-	12.28
Qualitative	1.34	*6.36*	*5.85*	*1.28*	*6.27*	**1.00**	-	**1.32**
The PAKDD	4.12	**1.51**	**1.15**	2.41	**1.41**	**1.00**	*2.00*	3.84

The RUS method obtained a perfectly balanced dataset, in all cases. CNN obtained good results for five datasets, but inverting the classes in two of them. It also has an interesting behavior for the Australian, Japanese and Qualitative datasets (original IR of 1.25, 1.26 and 1.34, respectively). In these datasets, the CNN method inverted the classes, and increase the imbalance ratio (to 3.37, 3.45 and 6.36, respectively). In the remaining datasets, CNN obtained IR from 1.12 to 2.62.

The CNNTL algorithm maintain the behavior of CNN in the Australian, Japanese and Qualitative datasets (returning IRs of 8.59, 9.07 and 5.85, respectively). In the remaining datasets, is had good performances (from 1.15 to 2.96), but inverting the classes in the Default credit, German and Give me credit datasets.

NCL algorithm obtained good results for the datasets having an original imbalance ratio lover than four, and it did not obtain a balanced dataset in the remaining ones. It also inverted the classes in the Australian, Japanese and Qualitative datasets.

The OSS algorithm had the same behavior of CNN and CNNTL in the Australian, Japanese and Qualitative datasets (with IRs of 5.50, 5.30 and 6.27, respectively). In the remaining datasets, it obtained good results (IRs from 1.21 to 2.56). Same as CNNTL, it inverted the classes in the Default credit, German and Give me credit datasets.

SBC algorithm had a disastrous behavior for the financial data. It inverted the classes in all datasets, and in nine cases it deleted the entire majority class (results marked with -). TL had good results for the almost balanced datasets (IR < 2), and did not obtained balanced results in the remaining data.

In general, the balancing methods evaluated but RUS had a poor performance. CNN, CNNTL and OSS obtained highly imbalanced results for almost balanced data (IR < 2). We consider that those methods should not be applied to balanced or near balanced data. For the remaining datasets, their results range from 1.12 to 2.62 (CNN), 1.15 to 2.96 (CNNTL) and 1.21 to 2.56 (OSS). On the other hand, the NCL algorithm showed good results for datasets having IR<4, and bad results for the remaining datasets.

In addition, all the compared algorithms but RUS inverted the classes in several datasets (converting the majority class into the minority one).

In the following, the results of the hybrid algorithms are offered in Table 8, for each of the datasets analyzed in the present investigation. The cases in which the classes were invested are shown in italics (the minority class became the majority), and in bold the good results (those with an imbalance closer to 1).

Table 8. Imbalance ratio for hybrid algorithms.

Datasets	SMOTE-ENN	SMOTE-TL	SPIDER	SPIDER2	Original
Australian	*1.03*	*1.31*	*1.17*	*1.49*	1.25
Default credit	*1.39*	*1.50*	**1.45**	**1.02**	3.52
German	*1.19*	*1.66*	**1.06**	*1.30*	2.33
Give me credit	*1.13*	*1.17*	4.22	2.62	13.96
Iranian	*1.18*	*1.14*	6.18	3.89	19.00
Japanese	*1.02*	*1.28*	*1.15*	*1.46*	1.26
Polish_1	*1.24*	*1.19*	6.87	4.14	24.93
Polish_2	*1.28*	*1.21*	6.49	3.84	24.43
Polish_3	*1.32*	*1.26*	5.50	3.31	20.22
Polish_4	*1.30*	*1.24*	5.08	3.13	18.01
Polish_5	*1.29*	*1.25*	3.98	2.51	13.41
Qualitative	**1.01**	*1.01*	**1.29**	**1.31**	1.34

The PAKDD	*1.33*	*1.16*	1.64	**1.05**	4.12

SMOTE-ENN and SMOTE-TL algorithms obtained very good balances in all cases (IRs from 1.01 to 1.66). SPIDER and SPIDER2 algorithms obtained good results for the datasets having an imbalance ratio lower than four, and bad results in the remaining datasets.

However, SPIDER2 algorithm obtained better results than SPIDER for the datasets having high imbalance (IR > 4). The IRs for SPIDER2 range from 2.51 to 4.14, while the ones of SPIDER range from 3.98 to 6.87.

In addition, we made a diagram summarizing some of the main characteristics of the compared methods (Figure 4). We include some of the positive and negative characteristics, according to the results obtained for instance sampling.

Algorithms	Characteristics			
	Positive		Negative	
	Obtains a balanced dataset	Deals with mixed and incomplete data	Increases data cardinality	Inverts majority/minority classes
SMOTE	X		X	
ADASYN	X		X	
SMOTE-BL	X		X	
SMOTE-SL	X		X	
ROS	X		X	
TL	If IR <4	X		
CNN		X		X
CNNTL		X		X
OSS				X
RUS	X	X		
NCL	If IR <4	X		
SBC				X
SMOTE-ENN	X			
SMOTE-TL	X			
SPIDER	X			
SPIDER2	X			

Figure 4. Characteristics of the compared methods.

Considering the experiments performed, we can conclude that some hybrid methods tend to obtain good balances; however, these results are at the expense in many cases, of inverting the majority class making it a minority. The next section will analyze how these results impact the performance of classifiers of the associative approach, and if this computational cost is justified in results of better performance.

5.2. Impact of the Sampling Algorithms in the Performance of Associative Classifiers

This section evaluates the impact of the results obtained by the different data balancing algorithms, in the performance of associative classifiers. For each of the datasets, after applying the balancing algorithms, the performance (Area under the ROC Curve) of four associative classifiers HACT, ED, NAC and SNDAM was calculated.

5.2.1. Impact of the Sampling Algorithms on the Performance of Associative Classifiers

The AUC results for the HACT classifier are presented in Tables 9–11 below. The good results (AUC improvements) are highlighted in bold, while the results that present less AUC than the original set (imbalanced) are underlined.

Table 9. AUC for HACT classifier after oversampling algorithms.

Datasets	Original	ADASYN	ADOMS	SMOTE-BL	ROS	SMOTE-SL	SMOTE
Australian	0.59	0.59	0.59	0.59	**0.61**	0.59	**0.60**
Default credit	0.95	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.97**
German	0.64	0.63	0.64	0.64	0.64	0.64	0.64
Give me credit	0.63	0.63	0.60	0.63	**0.64**	0.63	0.62
Iranian	0.61	0.60	0.61	0.60	0.61	0.60	0.60
Japanese	0.64	0.65	0.64	**0.66**	0.64	0.62	0.63
Polish_1	0.70	0.65	0.66	0.62	0.65	0.64	0.66
Polish_2	0.58	0.58	0.58	0.58	0.58	0.58	0.58
Polish_3	0.52	**0.53**	0.52	0.51	0.52	**0.53**	0.52
Polish_4	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Polish_5	0.53	0.52	0.52	0.53	0.52	0.52	0.52
Qualitative	0.55	0.55	0.55	0.55	0.55	**0.56**	0.55
The PAKDD	0.62	0.62	0.62	0.62	0.62	0.62	0.62

Table 10. AUC for HACT classifier after undersampling algorithms.

Datasets	Original	CNNTL	NCL	OSS	RUS	SBC	TL
Australian	0.59	0.58	0.59	0.57	0.59	**0.64**	0.59
Default credit	0.95	0.89	0.95	0.88	**0.96**	-	0.95
German	0.64	0.64	0.63	0.64	0.64	0.50	0.64
Give me credit	0.64	0.62	0.63	0.61	0.63	0.64	0.63
Iranian	0.61	0.61	0.61	0.61	0.61	-	0.61
Japanese	0.64	0.64	**0.65**	**0.67**	**0.65**	0.50	0.64
Polish_1	0.66	0.62	0.61	0.61	0.66	-	0.62
Polish_2	0.58	0.58	0.58	0.58	0.58	-	0.58
Polish_3	0.52	0.50	0.52	0.51	0.51	-	0.52
Polish_4	0.50	0.49	0.50	0.50	0.50	-	0.50
Polish_5	0.53	0.52	0.53	0.52	0.53	-	0.53
Qualitative	0.55	0.55	**0.56**	0.55	**0.56**	-	0.55
The PAKDD	0.62	**0.63**	0.62	**0.63**	0.62	0.53	**0.63**

Table 11. AUC for HACT classifier after hybrid algorithms.

Datasets	Original	SMOTE-ENN	SMOTE-TL	SPIDER2	SPIDER
Australian	0.59	**0.60**	0.57	0.59	0.59
Default credit	0.95	**0.96**	**0.96**	0.95	0.95
German	0.64	0.63	0.63	0.62	0.63
Give me credit	0.63	0.63	0.63	0.63	0.63
Iranian	0.61	0.61	0.61	0.61	0.61
Japanese	0.64	0.64	**0.65**	**0.65**	**0.65**
Polish_1	0.70	0.65	0.62	0.60	0.62
Polish_2	0.58	0.58	0.58	0.58	0.58
Polish_3	0.52	0.52	0.52	**0.53**	0.52
Polish_4	0.50	0.50	0.50	0.50	0.50
Polish_5	0.53	0.52	0.52	0.52	0.53
Qualitative	0.55	**0.56**	**0.56**	0.55	0.55
The PAKDD	0.62	0.62	0.62	0.62	**0.63**

The oversampling algorithms had slight drops and increases in the classifier performance, but no clear advantage was shown in the results. However, to determine if these differences in performance are significant or not, statistical tests were applied (Section 5.3).

A similar behavior was observed for undersampling algorithms, having slight drops and increases in the classifier performance, but with no clear advantages. Due to the SBC algorithm deleted the majority class in several datasets, its results in such data were not computed. Again, to determine if these differences in performance are significant or not, statistical tests were applied (Section 5.3).

As can be seen, the differences in performance for the HACT classifier were obtained in a few datasets, and never exceeded the original AUC by more than 2%. These results point to the inefficiency of the sampling algorithms for the improvement of this classifier.

Similarly, for the oversampling and hybrid algorithms, the slight improvements in performance in some datasets do not justify, in the opinion of the authors, the increase in the cardinality of the datasets.

5.2.2. Impact on the performance of the Extended Gamma Classifier

The AUC results for the Extended Gamma classifier are presented in Tables 12–14 below. The good results (AUC improvements) are highlighted in bold, while the results that present less AUC than the original set (imbalanced) are underlined.

Table 12. AUC for the Extended Gamma classifier after oversampling algorithms.

Datasets	Original	ADASYN	ADOMS	SMOTE-BL	ROS	SMOTE-SL	SMOTE
Australian	0.84	**0.85**	0.83	**0.85**	0.83	0.83	0.83
Default credit	0.99	0.98	0.99	0.97	0.99	0.99	0.99
German	0.67	0.63	0.61	0.64	0.67	0.67	0.64
Give me credit	0.68	0.67	0.54	0.67	**0.69**	**0.69**	0.65
Iranian	0.59	0.51	0.50	0.52	0.59	0.59	0.51
Japanese	0.68	0.63	0.57	0.58	0.67	0.64	0.63
Polish_1	0.82	**0.85**	**0.83**	**0.85**	0.82	0.82	0.82
Polish_2	0.57	0.55	0.50	**0.60**	**0.61**	**0.58**	**0.61**
Polish_3	0.76	0.50	0.50	0.50	0.76	0.58	0.50
Polish_4	0.71	0.50	0.50	0.50	0.71	0.55	0.50
Polish_5	0.71	0.50	0.50	0.50	**0.72**	0.62	0.50
Qualitative	0.75	0.50	0.50	0.50	0.75	0.67	0.50
The PAKDD	0.79	0.50	0.50	0.51	0.78	0.71	0.50

Table 13. AUC for the Extended Gamma classifier after undersampling algorithms.

Datasets	Original	CNNTL	NCL	OSS	RUS	SBC	TL
Australian	0.84	0.84	**0.87**	0.84	0.83	0.84	**0.85**
Default credit	0.99	0.94	0.99	0.94	0.99	-	0.99
German	0.67	**0.69**	0.66	**0.68**	0.67	0.53	0.66
Give me credit	0.68	0.68	0.68	**0.69**	0.68	0.67	**0.70**
Iranian	0.59	**0.60**	0.59	**0.60**	0.59	-	0.59
Japanese	0.68	**0.70**	0.68	**0.69**	0.67	0.50	0.68
Polish_1	0.82	**0.85**	**0.86**	**0.83**	0.82	-	**0.85**
Polish_2	0.61	0.60	0.60	0.61	0.61	-	0.61
Polish_3	0.76	0.76	0.75	0.74	0.75	-	0.76
Polish_4	0.71	0.68	0.71	0.69	0.70	-	0.71
Polish_5	0.71	**0.72**	0.71	**0.72**	0.71	-	0.71
Qualitative	0.75	0.73	0.75	0.73	0.74	-	0.75
The PAKDD	0.79	0.78	0.79	0.79	0.79	0.55	0.78

Table 14. AUC for Extended Gamma classifier after hybrid algorithms.

Datasets	Original	SMOTE-ENN	SMOTE-TL	SPIDER2	SPIDER
Australian	0.84	0.83	**0.85**	**0.86**	**0.85**
Default credit	0.99	**1.00**	**1.00**	0.98	0.98
German	0.67	0.65	0.65	0.65	0.65
Give me credit	0.68	**0.69**	**0.69**	0.68	0.68
Iranian	0.59	0.51	0.51	0.59	0.59
Japanese	0.68	0.63	0.63	0.67	0.67
Polish_1	0.82	0.82	**0.85**	**0.87**	**0.86**
Polish_2	0.57	0.56	0.57	**0.60**	**0.61**
Polish_3	0.76	0.50	0.50	0.76	0.76
Polish_4	0.71	0.50	0.50	0.71	0.71
Polish_5	0.71	0.50	0.50	0.71	0.72
Qualitative	0.75	0.50	0.50	0.75	0.74
The PAKDD	0.79	0.50	0.50	**0.80**	**0.80**

For the Extended Gamma classifier, the results of oversampling algorithms were similar to those obtained by the HACT classifier. There is no clear advantage of applying the oversampling algorithms, in the performance of the classifier.

For the undersampling algorithms, there is an improvement of classifier performance after applying OSS and CNNTL in six and five of the compared datasets, respectively. To determine if these differences in performance are significant or not, statistical tests were applied (Section 5.3).

For the Extended Gamma classifier, the hybrid algorithms showed a subtle improvement in performance in some datasets (ex. Australian, Default credit, Give me credit, Polish_1 and Polish_2). In the remaining of the datasets, differences can be seen in favor of the AUC are less than 1%. However, the SMOTE-ENN and SMOTE-TL algorithms had very unfavorable results in bankruptcy detection, evidencing a loss of more than 20% of AUC, with respect to the original.

Same as for the HACT classifier, in the case of the Extended Gamma classifier, for the oversampling algorithms, the slight improvements in performance obtained do not justify, in the authors' criteria, the increase in the cardinality of datasets.

5.2.3. Impact on the performance of the NAC Classifier

The AUC results for the NAC classifier are presented below, in Tables 15–17. The good results (AUC improvements) are highlighted in bold, while the results that present less AUC than the

original set (imbalanced) are underlined. The oversampling algorithms outperformed the results of using the original data in just five datasets. In all of them, the AUC improvements were of 1% only. Considering that oversampling algorithms increase the computational complexity by augmented the number of instances, we consider that its potential benefits for AUC do not justify its added complexity.

Table 15. AUC for NAC classifier after oversampling algorithms.

Datasets	Original	ADASYN	ADOMS	SMOTE-BL	ROS	SMOTE-SL	SMOTE
Australian	0.83	0.83	**0.84**	**0.84**	**0.84**	**0.84**	0.83
Default credit	0.99	0.98	**1.00**	0.98	0.99	0.99	0.99
German	0.67	0.65	0.67	0.67	0.67	0.66	0.67
Give me credit	0.69	0.69	0.68	0.68	**0.70**	0.68	0.69
Iranian	0.61	0.59	0.60	0.61	0.61	0.60	0.59
Japanese	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Polish_1	0.84	0.84	0.84	0.84	0.83	0.84	0.83
Polish_2	0.60	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**
Polish_3	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Polish_4	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Polish_5	0.50	0.50	0.50	0.50	**0.51**	0.50	0.50
Qualitative	0.53	0.51	0.52	0.50	0.53	0.50	0.51
The PAKDD	0.62	0.59	0.62	0.60	0.62	0.59	0.60

Table 16. AUC for NAC classifier after undersampling algorithms.

Datasets	Original	CNNTL	NCL	OSS	RUS	SBC	TL
Australian	0.83	**0.84**	0.83	0.83	**0.84**	0.82	0.83
Default credit	0.99	0.94	0.99	0.94	0.99	-	0.99
German	0.67	0.65	0.58	0.67	0.67	0.52	0.57
Give me credit	0.69	0.64	0.68	0.68	0.69	0.68	0.69
Iranian	0.61	0.53	**0.68**	0.54	0.61	-	**0.62**
Japanese	0.50	0.50	0.50	0.49	**0.54**	0.50	0.50
Polish_1	0.83	0.83	0.83	0.81	**0.85**	-	0.83
Polish_2	0.61	0.55	0.61	0.59	0.59	-	0.61
Polish_3	0.50	0.50	0.50	0.50	**0.52**	-	0.50
Polish_4	0.50	0.50	**0.52**	**0.51**	0.50	-	0.50
Polish_5	0.50	0.49	**0.55**	0.50	0.50	-	**0.53**
Qualitative	0.53	**0.54**	0.53	0.53	0.52	-	**0.54**
The PAKDD	0.62	0.50	**0.65**	0.52	0.50	0.50	0.57

Table 17. AUC for NAC classifier after hybrid algorithms.

Datasets	Original	SMOTE-ENN	SMOTE-TL	SPIDER2	SPIDER
Australian	0.83	**0.84**	0.82	0.82	0.83
Default credit	0.99	0.98	**1.00**	0.98	0.98
German	0.67	0.60	0.58	0.51	0.51
Give me credit	0.69	**0.70**	**0.70**	0.68	0.69
Iranian	0.61	**0.73**	**0.65**	0.61	**0.62**
Japanese	0.50	0.50	0.50	0.50	0.50
Polish_1	0.84	0.83	0.84	0.82	0.83
Polish_2	0.60	**0.61**	**0.61**	**0.61**	**0.61**
Polish_3	0.50	0.50	0.50	**0.51**	0.50
Polish_4	0.50	0.50	0.50	0.50	0.50
Polish_5	0.50	**0.53**	**0.52**	**0.51**	0.50

Qualitative	0.53	0.53	0.52	0.53	0.52
The PAKDD	0.62	0.53	0.56	0.61	0.61

According to undersampling algorithms, the best results were obtained by NCL and RUS, improving the AUC in four datasets. However, as it can be seen, on numerous occasions, using undersampling techniques negatively impacts the performance of the NAC classifier.

As can be seen in Table 17, the hybrid algorithms showed a slight improvement in the performance of the classifier in seven of the analyzed databases. The greatest improvement was obtained in the Iranian dataset, where the Area under the ROC Curve increased from 0.61 to 0.73, with the SMOTE-ENN algorithm. In the rest of the datasets analyzed, at least, strong falls were not observed in terms of classification performance.

For the NAC classifier, the data balancing algorithms did not show an obvious improvement over the original performance. It should be noted that the algorithms, in addition to involving a computational cost, in hybrid cases and oversampling, also increase the cardinality of the datasets.

5.2.4. Impact on the performance of the SNDAM Classifier

The AUC results for the SNDAM classifier are presented below, in Tables 18–20. The good results (AUC improvements) are highlighted in bold, while the results that present less AUC than the original set (imbalanced) are underlined.

Table 18. AUC for SNDAM classifier after oversampling algorithms.

Datasets	Original	ADASYN	ADOMS	SMOTE-BL	ROS	SMOTE-SL	SMOTE
Australian	0.80	0.79	0.79	**0.81**	0.80	0.80	0.80
Default credit	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**
German	0.61	**0.62**	0.61	**0.62**	0.61	0.61	**0.62**
Give me credit	0.58	**0.60**	**0.59**	**0.62**	0.58	**0.59**	0.58
Iranian	0.57	**0.58**	**0.59**	**0.59**	0.57	**0.58**	**0.59**
Japanese	0.65	**0.67**	**0.66**	**0.66**	0.65	0.65	**0.68**
Polish_1	0.82	0.82	**0.83**	**0.83**	0.82	0.82	0.82
Polish_2	0.91	0.91	0.91	0.90	0.91	0.78	0.91
Polish_3	0.54	**0.58**	**0.55**	**0.56**	0.54	0.54	**0.57**
Polish_4	0.52	0.52	**0.53**	0.52	0.52	0.52	**0.53**
Polish_5	0.52	**0.55**	**0.54**	**0.54**	0.52	**0.53**	**0.55**
Qualitative	0.54	**0.59**	**0.56**	**0.56**	0.54	0.54	**0.57**
The PAKDD	0.58	**0.62**	**0.60**	**0.61**	0.58	0.58	**0.61**

Table 19. AUC for SNDAM classifier after undersampling algorithms.

Datasets	Original	CNNTL	NCL	OSS	RUS	SBC	TL
Australian	0.80	0.71	**0.81**	0.77	**0.80**	**0.81**	**0.82**
Default credit	**1.00**	0.97	**1.00**	0.96	0.99	-	**1.00**
German	0.61	0.59	**0.65**	**0.62**	**0.62**	0.52	**0.64**
Give me credit	0.58	0.58	**0.63**	**0.59**	**0.59**	**0.62**	**0.61**
Iranian	0.57	0.57	**0.62**	**0.58**	**0.63**	-	**0.60**
Japanese	0.65	**0.72**	**0.68**	**0.68**	**0.70**	0.50	**0.67**
Polish_1	0.82	0.70	0.81	0.79	0.82	-	**0.84**
Polish_2	0.91	0.78	0.81	0.81	0.80	-	0.85
Polish_3	0.54	**0.56**	**0.56**	**0.58**	**0.65**	-	0.54
Polish_4	0.52	**0.55**	**0.53**	**0.55**	**0.60**	-	**0.53**
Polish_5	0.52	**0.58**	**0.54**	**0.56**	**0.60**	-	**0.53**
Qualitative	0.54	**0.57**	**0.58**	**0.57**	**0.62**	-	**0.56**
The PAKDD	0.58	**0.62**	**0.65**	**0.61**	**0.68**	0.53	**0.61**

Table 20. AUC for SNDAM classifier after hybrid algorithms.

Datasets	Original	SMOTE-ENN	SMOTE-TL	SPIDER2	SPIDER
Australian	0.80	**0.85**	**0.81**	**0.84**	**0.84**
Default credit	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
German	0.61	**0.64**	**0.64**	**0.64**	**0.64**
Give me credit	0.58	**0.64**	**0.64**	**0.61**	**0.60**
Iranian	0.57	**0.65**	**0.64**	**0.59**	**0.58**
Japanese	0.65	**0.66**	**0.68**	0.65	0.65
Polish_1	0.82	**0.87**	**0.84**	**0.85**	**0.85**
Polish_2	0.91	0.91	0.77	0.84	0.86
Polish_3	0.54	**0.59**	**0.60**	0.54	0.54
Polish_4	0.52	**0.54**	**0.54**	0.52	0.52
Polish_5	0.52	**0.55**	**0.55**	**0.53**	**0.53**
Qualitative	0.54	**0.62**	**0.61**	0.54	0.54
The PAKDD	0.58	**0.67**	**0.67**	**0.59**	**0.59**

Unlike the previously analyzed classifiers, the SNDAM showed an increase in AUC in 11 of the 13 compared datasets, after applying oversampling algorithms. These results point to the benefits of using such sampling techniques to increase SNDAM performance. However, to establish whether such AUC differences are significant or not, Section 5.3 uses statistical test.

In addition, undersampling algorithms also seem to increase the AUC of SNDAM, again for 11 of the 13 datasets. NCL, OSS, RUS and TL showed good results, although the increases in AUC were small (1%–4%) except for the Japanese dataset (7%).

The hybrid sampling algorithms obtained the best results, being able to increase SNDAM performance in 12 of the 13 datasets (SMOTE-ENN and SMOTE-TL) and in 8 datasets (SPIDER and SPIDER2). For the SNDAM classifier, the data balancing algorithms showed an evident improvement with respect to the original performance, unlike the other associative classifiers analyzed. Again, next section will address the statistical tests to determine if the differences in performance founded are significant or not.

5.3. Statistical Analysis

To establish whether the differences in the AUC founded in the previous section are significant or not, statistical tests were carried out.

The Friedman tests applied did not reject the hypothesis of equal performance when comparing the AUC of the balancing methods for the HACT, with p-values of 0.9799 for oversampling, 0.2116 for undersampling, and 0.9212 for hybrid algorithms. In this case, it is possible to conclude, with 95% certainty, that using class balancing methods DOES NOT improve the performance of HACT classifier, in imbalanced datasets, belonging to the financial field.

The test also did not reject the null hypothesis for oversampling and hybrid algorithms for NAC classifier (p-values of 0.2853 and 0.4980, respectively). For undersampling algorithms, the test did reject the null hypothesis, with a p-value of 0.0207. In the Friedman test, the best ranked algorithm was the original classifier, without instance sampling. For the undersampling algorithms, we applied the Holm's test (Table 21). Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.01.

Table 21. Results of the Holm test comparing the performance of the NAC classifier after undersampling algorithms.

i	Algorithm	z	p	Holm
6	SBC	2.768916	0.005624	0.008333
5	CNNTL	1.861075	0.062734	0.010000
4	OSS	1.770291	0.076679	0.012500
3	TL	0.408529	0.682886	0.016667
2	NCL	0.136176	0.891682	0.025000
1	RUS	0.045392	0.963795	0.050000

As shown by the test, we can conclude with a 95% of certainty, that the sampling algorithms but SBC, DID NOT improve the performance of the NAC classifier, in financial imbalanced data. In addition, SBC algorithm decreases the NAC performance.

Regarding the Extended Gamma classifier, the Friedman's tests did reject the null hypothesis of equal performance for oversampling, undersampling and hybrid algorithms. The corresponding p-values were 0.000048, 0.015356 and 0.013584, respectively. The best ranked algorithms were ROS, TL and SPIDER2. In these cases, post-hoc tests were performed to determine among which algorithms the differences were. Tables 22–24 show the results of the tests applied for oversampling, undersampling and hybrid methods, respectively.

Table 22. Results of the Holm test comparing the performance of the Extended Gamma classifier after oversampling algorithms. The test rejects the hypothesis having an unadjusted p-value ≤ 0.025.

i	Algorithm	z	p	Holm
6	ADOMS	3.994502	0.000065	0.008333
5	SMOTE	3.041268	0.002356	0.010000
4	ADASYN	2.995876	0.002737	0.012500
3	SMOTE-BL	2.496564	0.01254	0.016667
2	SMOTE-SL	0.998625	0.317976	0.025000
1	Original	0.136176	0.891682	0.050000

Table 23. Results of the Holm test comparing the performance of the Extended Gamma classifier after undersampling algorithms. The test rejects the hypothesis having an unadjusted p-value ≤ 0.01.

i	Algorithm	z	p	Holm
6	SBC	3.404405	0.000663	0.008333
5	RUS	1.361762	0.173273	0.010000
4	CNNTL	0.817057	0.413896	0.012500
3	OSS	0.635489	0.52511	0.016667
2	Original	0.499313	0.617559	0.025000
1	NCL	0.272352	0.785351	0.05

Table 24. Results of the Holm test comparing the performance of the Extended Gamma classifier after hybrid algorithms. The test rejects the hypothesis having an unadjusted p-value ≤ 0.016667.

i	Algorithm	z	p	Holm
4	SMOTE-ENN	2.790782	0.005258	0.012500
3	SMOTE-TL	2.10859	0.034980	0.016667
2	Original	0.496139	0.619796	0.025000
1	SPIDER	0.186052	0.852404	0.050000

The ROS algorithm had no significant differences with respect to the original classifier, nor with the SMOTE-SL algorithm. Then, we can conclude within a 95% confidence, that using oversampling

algorithms DID NOT increase the performance of the Extended Gamma classifier, using imbalanced financial data.

The best ranked undersampling algorithm, TL, had no significant difference in performance with none of the remaining undersampling algorithms but SBC, nor with respect using the original imbalanced dataset. The SBC algorithm was indeed significantly worse than TL, for the Extended Gamma Classifier performance. Again, we can conclude within a 95% confidence, that using undersampling algorithms DID NOT increase the performance of the Extended Gamma classifier, using imbalanced financial data.

The best ranked hybrid algorithm SPIDER2, had no significant difference in performance with SPIDER and SMOTE-TL nor with respect using the original imbalanced dataset. The SMOTE-ENN algorithm was significantly worse than SPIDER2, for the Extended Gamma classifier performance. Again, we can conclude within a 95% confidence, that using hybrid algorithms DID NOT increase the performance of the Extended Gamma classifier, using imbalanced financial data.

With respect the SNDAM classifier, the best ranked algorithms according to the Friedman tests were SMOTE, RUS and SMOTE-ENN. The results of the corresponding Holm's test for oversampling, undersampling and hybrid algorithms are showed in Tables 25–27.

Table 25. Results of the Holm test comparing the performance of the SNDAM classifier after oversampling algorithms. The test rejects the hypothesis having an unadjusted p-value ≤ 0.016667.

i	Algorithm	z	p	Holm
6	Original	2.995876	0.002737	0.008333
5	ROS	2.995876	0.002737	0.010000
4	SMOTE-SL	2.814308	0.004888	0.012500
3	ADOMS	1.225586	0.220355	0.016667
2	ADASYN	0.22696	0.820455	0.025000
1	SMOTE-BL	0.22696	0.820455	0.050000

Table 26. Results of the Holm test comparing the performance of the SNDAM classifier after undersampling algorithms. The test rejects the hypothesis having an unadjusted p-value ≤ 0.01.

i	Algorithm	z	p	Holm
6	SBC	2.950484	0.003173	0.008333
5	CNNTL	2.314995	0.020613	0.010000
4	Original	2.088035	0.036795	0.012500
3	OSS	1.679506	0.093053	0.016667
2	TL	0.726273	0.467671	0.025000
1	NCL	0.090784	0.927664	0.050000

Table 27. Results of the Holm test comparing the performance of the SNDAM classifier after hybrid algorithms. The test rejects the hypothesis having an unadjusted p-value≤0.05.

i	Algorithm	z	p	Holm
4	Original	4.279198	0.000019	0.012500
3	SPIDER	2.914816	0.003559	0.016667
2	SPIDER2	2.790782	0.005258	0.025000
1	SMOTE-TL	1.17833	0.238665	0.050000

The SMOTE algorithm did not have significant differences with respect the SMOTE-BL, ADASYN and ADOMS algorithms, due to the null hypothesis were not rejected for such cases. However, SMOTE showed a significantly better performance than SMOTE-SL, ROS and the original classifier without instance selection. The statistical tests allow us to state that using oversampling techniques such as SMOTE, DID increase the Area under the ROC curve of the SNDAM classifier,

over imbalanced financial data. Such improvement came with the additional computational cost of creating artificial instances and therefore increasing the cardinality of the datasets.

The best ranked undersampling algorithm, RUS, had no significant difference in performance with none of the remaining undersampling algorithms but SBC, nor with respect using the original imbalanced dataset. Again, we can conclude within a 95% confidence, that using undersampling algorithms DID NOT increase the performance of the SNDAM classifier, using imbalanced financial data.

The Holm's test did not found significant differences in performance between the SMOTE-ENN and SMOTE-TL algorithms. However, the test did found SMOTE-ENN being significantly better than SPIDER2, SPIDER and the original classifier without instance sampling. Then, can conclude within a 95% confidence, that using oversampling algorithms such as SMOTE-ENN DID increase the performance of the SNDAM classifier, using imbalanced financial data.

From the experiments performed, it is possible to establish that the HACT, Extended Gamma and NAC classifiers DO NOT benefit from data balancing. On the other hand, the SNDAM classifier DOES obtain improvements in its performance by balancing the datasets using oversampling (as SMOTE) and hybrid (as SMOTE-ENN) algorithms. Undersampling algorithms DO NOT improve the performance of the SNDAM classifier over imbalanced financial data.

6. Conclusions and Future Works

In this paper, an in-depth study was carried out on data balancing techniques, as well as their application in financial data, and their impact in the performance of associative classifiers. This study allowed us to reach the following conclusions:

1. About sampling methods:
    a. All of the oversampling methods tested obtained balanced datasets, although at the cost of increasing the cardinality of the data.
    b. The undersampling methods analyzed (CNN, CNNTL, NCL, OSS, SBC and TL), but RUS, fail to find balanced data sets, when the imbalance ratio of the original set was greater than 4.0. However, for moderate imbalance ratios (less than 4.0), the NCL and TL algorithms got good results.
    c. Systematically, the CNN, CNNTL, OSS and SBC algorithms reversed the amounts of instances in the classes in the datasets, making the majority class a minority.
    d. The SBC algorithm had a very bad behavior in the face of financial data, since it systematically eliminated all the instances of the majority class.
    e. Both SMOTE-ENN and SMOTE-TL obtained good results according to data balancing.
    f. SPIDER2 obtained better balanced datasets than SPIDER.
2. About the impact of the sampling in the associative classifiers:
    a. The HACT, Extended Gamma and NAC classifiers do not benefit from financial data balancing.
    b. Undersampling algorithms do not benefit the SNDAM classifier. However, oversampling and hybrid methods do increase the performance of SNDAM over imbalanced financial data.
    c. There is a significant improvement, within a 95% of confidence, in the Area under the ROC curve of SNDAM while sampling imbalanced financial data by SMOTE and SMOTE-ENN.

Considering the above, it is proposed as future work of the research:

1. To design undersampling algorithms that are robust to high imbalance ratios, in order to solve the limitations founded in the evaluated algorithms.
2. To apply the proposed methodology to other supervised classifiers, for instance Deep Neural Networks and other algorithms related with Deep Learning.

3. To choose other datasets, related to areas of interest other than financial, in order to perform experiments similar to those presented in this paper.

Author Contributions: Conceptualization, C.Y.-M. and Y.V-R.; methodology, Y.V.-R.; software, A.R.-D.d.-l.V.; validation, O.C.-N., formal analysis, Y.V.-R. and C.Y.-M.; investigation, I.L.-Y.; writing—original draft preparation, Y.V.-R.; writing—review and editing, C.Y.-M.; visualization, A.R.-D.d.-l.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors gratefully acknowledge the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, CIC and CIDETEC), the Consejo Nacional de Ciencia y Tecnología (Conacyt), and Sistema Nacional de Investigadores for their economic support to develop this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bischl, B.; Kühn, T.; Szepannek, G. On Class Imbalance Correction for Classification Algorithms in Credit Scoring. In *Operations Research Proceedings 2014*; Springer: Basel, Switzerland, 2014; pp. 37–43.
2. García, V.; Marqués, A.; Sánchez, J.S. On the use of data filtering techniques for credit risk prediction with instance-based models. *Expert Syst. Appl.* **2012**, *39*, 13267–13276.
3. Marqués, A.I.; García, V.; Sánchez, J.S. On the suitability of resampling techniques for the class imbalance problem in credit scoring. *J. Oper. Res. Soc.* **2013**, *64*, 1060–1070.
4. Banasik, J.; Crook, J.; Thomas, L. Sample selection bias in credit scoring models. *J. Oper. Res. Soc.* **2003**, *54*, 822–832.
5. Su, H.; Qi, W.; Yang, C.; Aliverti, A.; Ferrigno, G.; De Momi, E. Deep Neural Network Approach in Human-Like Redundancy Optimization for Anthropomorphic Manipulators. *IEEE Access* **2019**, *7*, 124207–124216.
6. Su, H.; Yang, C.; Mdeihly, H.; Rizzo, A.; Ferrigno, G.; De Momi, E. Neural Network Enhanced Robot Tool Identification and Calibration for Bilateral Teleoperation. *IEEE Access* **2019**, *7*, 122041–122051.
7. Goh, R.; Lee, L. Credit Scoring: A Review on Support Vector Machines and Metaheuristic Approaches. *Adv. Oper. Res.* **2019**, *2019*, 1–30.
8. Wang, T.; Li, J. An improved support vector machine and its application in P2P lending personal credit scoring. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; p. 062041.
9. Luo, J.; Yan, X.; Tian, Y. Unsupervised quadratic surface support vector machine with application to credit risk assessment. *Eur. J. Oper. Res.* **2020**, *280*, 1008–1017.
10. Akkoç, S. Exploring the Nature of Credit Scoring: A Neuro Fuzzy Approach. *Fuzzy Econ. Rev.* **2019**, *24*, 3–24.
11. Livieris, I.E. Forecasting economy-related data utilizing weight-constrained recurrent neural networks. *Algorithms* **2019**, *12*, 85.
12. Munkhdalai, L.; Lee, J.Y.; Ryu, K.H. A Hybrid Credit Scoring Model Using Neural Networks and Logistic Regression. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*; Springer: Basel, Switzerland, 2020; pp. 251-258.
13. Feng, X.; Xiao, Z.; Zhong, B.; Dong, Y.; Qiu, J. Dynamic weighted ensemble classification for credit scoring using Markov Chain. *Appl. Intell.* **2019**, *49*, 555–568.
14. Guo, S.; He, H.; Huang, X. A multi-stage self-adaptive classifier ensemble model with application in credit scoring. *IEEE Access* **2019**, *7*, 78549–78559.
15. Pławiak, P.; Abdar, M.; Acharya, U.R. Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring. *Appl. Soft Comput.* **2019**, *84*, 105740.
16. Xiao, J.; Zhou, X.; Zhong, Y.; Xie, L.; Gu, X.; Liu, D. Cost-sensitive semi-supervised selective ensemble model for customer credit scoring. *Knowl.-Based Syst.* **2020**, *189*, 105118.
17. Shen, K.-Y.; Sakai, H.; Tzeng, G.-H. Comparing two novel hybrid MRDM approaches to consumer credit scoring under uncertainty and fuzzy judgments. *Int. J. Fuzzy Syst.* **2019**, *21*, 194–212.
18. Zhang, W.; He, H.; Zhang, S. A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Syst. Appl.* **2019**, *121*, 221–232.

19. Maldonado, S.; Peters, G.; Weber, R. Credit scoring using three-way decisions with probabilistic rough sets. *Inf. Sci.* **2020**, *507*, 700–714.
20. García, V.; Marqués, A.I.; Sánchez, J.S. Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. *Inf. Fusion* **2019**, *47*, 88–101.
21. Louzada, F.; Ara, A.; Fernandes, G.B. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surv. Oper. Res. Manag. Sci.* **2016**, *21*, 117–134.
22. Lessmann, S.; Baesens, B.; Seow, H.-V.; Thomas, L.C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *Eur. J. Oper. Res.* **2015**, *247*, 124–136.
23. Brown, I.; Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.* **2012**, *39*, 3446–3453.
24. Su, H.; Yang, C.; Ferrigno, G.; De Momi, E. Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1447–1453.
25. Wolpert, D.H. The supervised learning no-free-lunch theorems. In *Soft Computing and Industry*; Springer: London, UK, 2002; pp. 25–42.
26. Villuendas-Rey, Y.; Rey-Benguría, C.F.; Ferreira-Santiago, Á.; Camacho-Nieto, O.; Yáñez-Márquez, C. The naïve associative classifier (NAC): A novel, simple, transparent, and accurate classification model evaluated on financial data. *Neurocomputing* **2017**, *265*, 105–115.
27. López, V.; Fernández, A.; García, S.; Palade, V.; Herrera, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **2013**, *250*, 113–141.
28. Piramuthu, S. On preprocessing data for financial credit risk evaluation. *Expert Syst. Appl.* **2006**, *30*, 489–497.
29. Abdou, H.A.; Pointon, J. Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intell. Syst. Account. Financ. Manag.* **2011**, *18*, 59–88.
30. Su, H.; Ovur, S.E.; Zhou, X.; Qi, W.; Ferrigno, G.; De Momi, E. Depth vision guided hand gesture recognition using electromyographic signals. *Adv. Robot.* **2020**, 1–13. doi:10.1080/01691864.2020.1713886.
31. Beaver, W.H. Financial ratios as predictors of failure. *J. Account. Res.* **1966**, *4*, 71–111.
32. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Financ.* **1968**, *23*, 589–609.
33. Damrongsakmethee, T.; Neagoe, V.-E. Principal component analysis and relieff cascaded with decision tree for credit scoring. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 24–27 April 2019; pp. 85–95.
34. Kozodoi, N.; Lessmann, S.; Papakonstantinou, K.; Gatsoulis, Y.; Baesens, B. A multi-objective approach for profit-driven feature selection in credit scoring. *Decis. Support Syst.* **2019**, *120*, 106–117.
35. Srinivasan, V.; Kim, Y.H. Credit granting: A comparative analysis of classification procedures. *J. Financ.* **1987**, *42*, 665–681.
36. Abellán, J.; Castellano, J.G. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Syst. Appl.* **2017**, *73*, 1–10.
37. Boughaci, D.; Alkhawaldeh, A.A. Appropriate machine learning techniques for credit scoring and bankruptcy prediction in banking and finance: A comparative study. *Risk Decis. Anal.* **2018**, 1–10. doi:10.3233/RDA-180051.
38. Greene, W. Sample selection in credit-scoring models. *Jpn. World Econ.* **1998**, *10*, 299–316.
39. Crone, S.F.; Finlay, S. Instance sampling in credit scoring: An empirical study of sample size and balancing. *Int. J. Forecast.* **2012**, *28*, 224–238.
40. Dal Pozzolo, A.; Caelen, O.; Bontempi, G. When is undersampling effective in unbalanced classification tasks? In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Porto, Portugal, 7–11 September 2015, pp. 200-215.
41. Santiago-Montero, R. Hybrid Accociative pattern Classifier with Translation (In Spanish: Clasificador Híbrido de Patrones Basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson Kohonen). Master's Thesis, Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, Mexico, 2003.
42. Cleofas-Sánchez, L.; García, V.; Marqués, A.; Sánchez, J.S. Financial distress prediction using the hybrid associative memory with translation. *Appl. Soft Comput.* **2016**, *44*, 144–152.
43. López-Yáñez, I.; Argüelles-Cruz, A.J.; Camacho-Nieto, O.; Yáñez-Márquez, C. Pollutants time-series prediction using the Gamma classifier. *Int. J. Comput. Int. Syst.* **2011**, *4*, 680–711.

44. Ramirez, A.; Lopez, I.; Villuendas, Y.; Yanez, C. Evolutive improvement of parameters in an associative classifier. *IEEE Lat. Am. Trans.* **2015**, *13*, 1550–1555.
45. Villuendas-Rey, Y.; Yanez-Marquez, C.; Anton-Vargas, J.A.; Lopez-Yanez, I. An extension of the gamma associative classifier for dealing with hybrid data. *IEEE Access* **2019**, *7*, 64198–64205.
46. Serrano-Silva, Y.O.; Villuendas-Rey, Y.; Yáñez-Márquez, C. Automatic feature weighting for improving financial Decision Support Systems. *Decis. Support Syst.* **2018**, *107*, 78–87.
47. Ramírez-Rubio, R.; Aldape-Pérez, M.; Yáñez-Márquez, C.; López-Yáñez, I.; Camacho-Nieto, O. Pattern classification using smallest normalized difference associative memory. *Pattern Recogn. Lett.* **2017**, *93*, 104–112.
48. Cleofas-Sánchez, L.; Sánchez, J.S.; García, V.; Valdovinos, R.M. Associative Learning on imbalanced environments: An empirical study. *Expert Syst. Appl.* **2016**, *54*, 387–397.
49. González, S.; García, S.; Li, S.-T.; Herrera, F. Chain based sampling for monotonic imbalanced classification. *Inf. Sci.* **2019**, *474*, 187–204.
50. Nejatian, S.; Parvin, H.; Faraji, E. Using sub-sampling and ensemble clustering techniques to improve performance of imbalanced classification. *Neurocomputing* **2018**, *276*, 55–66.
51. Yan, Y.; Liu, R.; Ding, Z.; Du, X.; Chen, J.; Zhang, Y. A parameter-free cleaning method for SMOTE in imbalanced classification. *IEEE Access* **2019**, *7*, 23537–23548.
52. Li, Y.; Wang, J.; Wang, S.; Liang, J.; Li, J. Local dense mixed region cutting+ global rebalancing: A method for imbalanced text sentiment classification. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1805–1820.
53. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.
54. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
55. Han, H.; Wang, W.-Y.; Mao, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Proceedings of the International Conference on Intelligent Computing, Hefei, China, 23–26 August 2005; pp. 878–887.
56. Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Bangkok, Thailand, 27–30 April 2009; pp. 475–482.
57. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29.
58. Tang, S.; Chen, S.-P. The generation mechanism of synthetic minority class examples. In Proceedings of the 2008 International Conference on Information Technology and Applications in Biomedicine, Shenzhen, China, 30–31 May 2008; pp. 444–447.
59. Tomek, I. Two modification of CNN. *IEEE Trans. Syst. Man Commun.* **1976**, *6*, 769–772.
60. Hart, P. The condensed nearest neighbor rule. *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516.
61. Kubat, M.; Matwin, S. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of 14th International conference on Machine Learning (ICML)*, ICML: Nashville, TN, USA; 1997; pp. 179–186.
62. Laurikkala, J. Improving identification of difficult small classes by balancing class distribution. In Proceedings of the Conference on Artificial Intelligence in Medicine in Europe, Cascais, Portugal, 1–4 July 2001; pp. 63–66.
63. Yen, S.-J.; Lee, Y.-S. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*; Springer: Berlin, Heidelberg, 2006; pp. 731–740.
64. Stefanowski, J.; Wilk, S. Selective pre-processing of imbalanced data for improving classification performance. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Turin, Italy, 1–5 September 2008; pp. 283–292.
65. Napierała, K.; Stefanowski, J.; Wilk, S. Learning from imbalanced data in presence of noisy and borderline examples. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Warsaw, Poland, 28–30 June 2010; pp. 158–167.
66. Larson, S.C. The shrinkage of the coefficient of multiple correlation. *J. Educ. Psychol.* **1931**, *22*, 45–55.
67. Stone, M. Cross-Validatory Choice and Assessment of Statistical Predictions. *J. R. Stat. Soc.* **1974**, *36*, 111–147.

68. Geisser, S. The predictive sample reuse model method with applications. *J. Am. Stat. Assoc.* **1975**, *70*, 320–328.
69. Dietterich, T.G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput.* **1998**, *10*, 1895–1923.
70. Alcalá-Fdez, F.H.J.; Sánchez, L.; García, S.; del Jesus, M.J.; Ventura, S.; Garrell, J.M.; Otero, J.; Romero, C.; Bacardit, J.; Rivas, V.M.; et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput., **2009**, *13*, 307–318.
71. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437.
72. Triguero, I.; González, S.; Moyano, J.M.; García López, S.; Alcalá Fernández, J.; Luengo Martín, J.; Fernández, A.; del Jesús, M.J.; Sánchez, L.; Herrera, F. KEEL 3.0: An open source software for multi-stage analysis in data mining. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 1238–1249.
73. Hernández-Castaño, J.A.; Villuendas-Rey, Y.; Camacho-Nieto, O.; Yáñez-Márquez, C. Experimental platform for intelligent computing (epic). *Computación y Sistemas* **2018**, *22*, 245–253.
74. Hernández-Castaño, J.A.; Villuendas-Rey, Y.; Camacho-Nieto, O.; Rey-Benguría, C.F. A New Experimentation Module for the EPIC Software. *Res. Comput. Sci.* **2018**, *147*, 243–252.
75. Garcia, S.; Herrera, F. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
76. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701.
77. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.